

# Suivi des commandes des revendeurs

## Contexte professionnel

L'entreprise **Distributech**, grossiste en équipements électroniques, collabore avec un réseau de **revendeurs régionaux**. Ces revendeurs passent régulièrement **des commandes** pour réapprovisionner leurs stocks. L'entreprise souhaite mettre en place une **base de données centrale** pour :

- Suivre l'historique des **commandes passées** (depuis des fichiers CSV envoyés par les revendeurs),
- Conserver une **photographie des stocks** dans chaque région (issus d'une base SQLite mise à jour par les commerciaux).

Un **processus ETL hebdomadaire** doit extraire, transformer et charger les données dans une **base SQL** pour permettre un meilleur suivi logistique et commercial.

---

## Objectifs pédagogiques

Les étudiants apprendront à :

- Créer une base SQL relationnelle structurée autour de revendeurs, régions, produits, commandes et stocks,
  - Implémenter un **pipeline ETL en Python** pour intégrer des données depuis une base **SQLite** et des **fichiers CSV**,
  - Modéliser et insérer des **stocks historiques** pour suivre leur évolution dans le temps.
- 

## Sources de données

- ♦ **Fichiers CSV (1 par semaine)**

Chaque revendeur envoie un fichier listant ses **commandes de produits** :

- ♦ **Base SQLite locale**

Les stocks de Distributech sont dans une base SQLite.

Voici les **règles de gestion** des **revendeurs**, **commandes** et **stocks** pour le projet :

---

### **Revendeurs**

- Chaque revendeur est associé à une **région unique**.
  - Un **identifiant unique** permet de les distinguer.
  - Les données des revendeurs sont extraites depuis la base **SQLite**.
- 

## Produits

- Chaque produit possède un **identifiant unique** et un **nom**.
  - Chaque produit possède un **coût unitaire** (ce que paie le revendeur).
  - Les produits sont communs à tous les revendeurs (catalogue unique).
- 



## Commandes

- Chaque commande a un **numéro de commande unique**.
  - Une commande est liée à un **revendeur** et à sa **région**.
  - Une commande possède une **date**.
  - Une commande est composée d'**une ou plusieurs lignes de commande**.
  - Chaque ligne de commande contient un **produit**, une **quantité**, et un **prix unitaire**.
  - Le fichier CSV contient les commandes passées par les revendeurs.
- 



## Stocks

- Le stock est **centralisé** : tous les revendeurs partagent le **même stock global**.
  - Le stock est mis à jour via deux **types d'événements** :
    - Les **réceptions** de produits réapprovisionnements dans la base SQLite,
    - Les **commandes** passées par les revendeurs par CSV
  - Chaque événement de stock est enregistré dans la table avec :
    - La **date**,
    - Le **produit**,
    - La **quantité**,
  - Le stock courant peut être reconstitué à date, par une soustraction des commandes du stock général.
- 

## Base de données cible (SQL à créer via script Python)

### La base doit :

1. identifie les zones géographiques
2. Les revendeurs de Distributech
3. Le catalogue de produits
4. Enregistre chaque commande passée par un revendeur

5. Trace l'évolution des niveaux de stock par produit et revendeur
- 

## Tâches à réaliser

### 1. Création de la base SQL

### 2. Développement du processus ETL

- **Extract :**
    - Lit les commandes depuis un fichier CSV (plusieurs fichiers possibles),
    - Lit la base SQLite pour obtenir revendeurs, régions, et stocks actuels.
  - **Transform :**
    - Vérifie la cohérence des données,
    - Supprime les doublons, nettoie les formats de date, etc.
  - **Load :**
    - Mettre à jour la base SQL,
    - Générer un CSV de l'état des stocks par produit.
- 

## Livrables attendus

- Scripts ETL.
  - Un fichier d'export SQL de la base complète.
  - Un CDC et des spécifications.
  - Le csv de l'état des stocks.
-